functions labeling the leaves reachable from one of the nodes corresponds to the set of functions which depend on the variables labeling the one of the nodes,traverses the labels in a selected order and sifts the variables of the binary decision diagram based on the selected order; and

a second store, wherein the sifted variables of the binary decision diagram are written by the processor to the second store.

9. (Amended) The apparatus of claim 8, wherein the number is a threshold derived from an original number of nodes.

10. (Amended) Apparatus as claimed in claim 8, wherein the number is a number of nodes which branch on a predetermined variable.

11. (Amended) The apparatus of claim 8, wherein the number is an absolute number.

REMARKS

In response to the Office Action mailed June 6, 2002, Applicant respectfully requests reconsideration. Claims 1-11 are pending in this application, of which claims 1-3 and 6-8 are independent.

Amendments to the Specification:

Applicant has amended the specification by presenting a substitute specification, pursuant to 37 C.F.R. §§1.121(b)(3) and 1.125(b), in which errors and informalities, as pointed out in the Office Action at paragraph 4, have been corrected and addressed including those in the title and Abstract.

Furthermore, a paragraph has been added to the Detailed Description describing that which is shown in newly-presented Figure 8. The subject matter of Figure 8 and the corresponding description was set forth in the original application and claims, for example at p. 2, middle – p. 5, middle of the originally-filed specification and in originally-filed claims 1-8. Accordingly, no new matter is being presented by introduction of new Figure 8 and the corresponding description. Also see Applicant's remarks below with respect to the drawings.

Amendments to the Drawings:

At paragraph 6, the Office Action objected to Figures 1-4 as not being labeled "Prior Art." Figures 1-4 have been designated in the legend as --Prior Art--, as shown in red on the accompanying drawings. The amended figures now conform to the statutory requirements and 37 C.F.R. §1.83(a).

Additionally, the Office Action objected to the drawings under 37 C.F.R. §1.83(a), at paragraph 5, stating that the drawings do not show every feature of the invention specified in the claims. Applicant hereby presents new Figure 8, whose introduction adds no new matter.

The specific limitations cited in the Examiner's objection were:

"adapted to arrange the variables,"

"arranging the variables,"

"labeling the nodes,"

"produce a list for said labels,"

"substituting functions which determine the values of internal signals into the set of functions representing said system,"

"threshold derived from an original number [of] nodes,"

"sifting in reverse order,"

"selected order to a deepest best location followed by sifting in reverse order to a shallowest best location,"

"detecting an increase in the number of nodes of said binary decision diagram," and

"in response to such detection arranging the variable of said binary decision diagram on the nodes of a graph."

Applicant presents new Figure 8 in which all of the above limitations and elements are shown or recited. A paragraph closely paralleling the content of new Figure 8 has been added to the specification. The above-mentioned limitations were all drawn from the originally-filed claims and find support in the original specification, e.g. at p. 2, middle – p. 5, middle. Accordingly, no new matter has been introduced by the addition of new Figure 8 and the corresponding detailed description of Figure 8.

Applicant respectfully requests examination and approval of the new and amended figures.

Regarding the Objections to the Claims:

At paragraph 8 of the Office Action, claims 1, 3, 6-7 were objected to for various informalities and minor grammatical errors. Applicant has corrected or amended each of the claims with respect to the Examiner's objections under 37 C.F.R. §1.75(a) and (d). Accordingly, Applicant respectfully requests that the Examiner withdraw the objections under 37 C.F.R. §1.75(a) and (d) to these claims.

The Office Action objected to claims 1-11 for containing unclear language. Office Action at paragraph 9. Applicant has reviewed the claims and has amended the claims to clarify and correct any ambiguities in the claims. Applicant now believes that all claims are in accordance with 37 C.F.R. §1.75(a) and (d), as well as 35 U.S.C. §112. Applicant respectfully requests that the objections made at paragraph 9 of the Office Action now be withdrawn.

Regarding the objection made at paragraph 9(c), Applicant respectfully disagrees with the Examiner's interpretation of the phrase "adapted to" as being interpreted to be "programmed or alternately hard wired." Applicant maintains that "adapted to" encompasses at least programmed and hardwired embodiments. The phrase "adapted to" is not counter to any statutory requirement or rule, and Applicant respectfully requests that the claim language be interpreted as recited and not through any other interpretation of the claim language.

Similarly, the Examiner interprets the Applicant's claims at paragraph 10 the Office Action. Applicant respectfully requests that the Examiner be guided by that which is recited in the claims rather than formulating alternate characterizations of Applicant's claims. Applicant directs the Examiner to the specification and the claims for the correct interpretation of the claims and disagrees with substitute interpretations made in the Office Action, regardless of their merit or correctness.

Regarding Rejections of the Claims (Generally):

To further the prosecution of this application, clarifying amendments have been made in the claims, as illustrated in the document submitted herewith entitled "Marked-up Claims." Applicant points out that the amendments made to the claims have been made to further clarify the claimed subject matter and not in an effort to distinguish over the cited art, as that is believed to be unnecessary for at least the reasons identified below.

Claim Rejections under 35 U.S.C. §112

Claims 1-11 were rejected under 35 U.S.C. §112, first paragraph, as containing subject matter which was not described in the specification in such a way as to enable one skilled in the art to make and/or use the invention. Several limitations were recited in the Office Action at paragraph 12 in this regard.

Applicant respectfully traverses this rejection and presents amended claims 1-11, which further clarify that which is being claimed. Applicant respectfully requests that this rejection now be withdrawn.

Specifically, the first three features recited in the Office Action "adapted to arrange the variables," "arranging the variables" and "labeling the nodes," relate to the construction of a function graph. An embodiment of a function graph is illustrated in the present application in Figure 6. A schematic diagram of an embodiment of the apparatus used in constructing the function graph as shown in Figure 7 of the present application. Exemplary rules for the construction of a function graph are discussed in the description of the original specification at p. 8, 5th paragraph. It is well-known to those skilled in the art how to adapt a processor to arrange numerical information to represent graphical information, and hence, a more detailed description is not required.

The limitation of "producing a list of said labels" relates to traversing the function graph in order to generate a list of variables in a selected order. A description of this is given in the original specification at p. 8, last line - p. 9, line 16. Accordingly, it is submitted that these features are sufficiently described in the present application to enable one skilled in the art to make and/or use the invention without undue experimentation.

The limitation of "sifting in reverse order" and "selected order to a deepest best location followed by sifting in reverse order to a shallowest best location" relate to sifting the variables. Sifting is described in the original specification at p. 8, second and third paragraphs and at p. 9, lines 18-21. Furthermore, sifting methods have been described in the related art, for example in Ashar and in Rudell, both of which are cited by the Examiner in the Office Action. Although the present method differs from the cited prior art, the terminology used is well-known to those skilled in the art, as evidenced by Ashar and Rudell, and thus a more detailed explanation is not required.

The limitation of "detecting an increase in the number of nodes of said binary decision diagram," "in response to such detection arranging the variables of said binary decision diagram on the nodes of a graph" "said number of a threshold derived from an original number of nodes" and "substituting functions which determine the values of internal signals into the set of functions representing said system" are also sufficiently described and enabled in the present application. Detection of an increase in the number of nodes is described, e.g., at p. 1 of the specification, which discusses the possibility of an explosion in the size of the BDD representation, or an increase in the number of nodes. Also discussed is the need to detect such an increase and suspend the substitution process while the representation is restructured in order to reduce its size. The feature of substitution of functions which determine the values of internal signals is discussed at the last two paragraphs of p. 1 of the original specification as well as on p. 7 of the original specification, now found at pp. 1 and 6-7 of the substitute specification. The substitution processes are described in the prior art section of the specification, and are generally known to those skilled in the art. Since one of skill in the art will be familiar with the prior art and the process of substitution in general, the novel aspect further described by the present application, e.g., statically ordering the variables using a graphical technique to select an ordered list, will be understood by that person skilled in the art. Furthermore, the addition of new Figure 8, and the associated description clearly illustrates that which is being claimed to enable those skilled in the art to make and/or use the invention without undue experimentation.

Claims 4-6 and 8-11 were rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. Applicant respectfully traverses this rejection and presents amended claims 4-6 and 8-11, which more clearly point out and claim the subject matter which applicant believes to be the invention.

With regard to the use of the use of the term "best," the use is justified by that which is known to those skilled in the art as well as by the specification which clearly illustrates the meaning of the term best in the present context. For example, the specification (original at p. 2, lines 13-15, now at p. 2, lines 7-9) recites that in BDDs, "better orderings result in fewer nodes in the graph." Hence, one of skill in the art would appreciate that a "best" ordering corresponds to an ordering resulting in the fewest nodes in the graph.

Claim Rejections Under 35 U.S.C. §101

The Office Action rejected claims 1-2 under 35 U.S.C. §101 as non-statutory subject matter. Office Action paragraph 15. While Applicant disputes that the claims as originally filed are not directed to statutory subject matter, the claims have been amended to clearly be directed to statutory subject matter. Thus, the claim rejections under 35 U.S.C. §101 are now moot in light of a present amendment, including amendments to claims 1 and 2. Accordingly, Applicant respectfully requests that this rejection be withdrawn.

The Examiner states, at paragraph 15 of the Office Action that "claims 1 and 2 merely recite mathematical algorithms" as a grounds for rejection under 35 U.S.C. §101. Applicant respectfully draws the Examiner's attention to the MPEP §2106(II) that states "Office personnel will no longer begin examination by determining if a claim recites a 'mathematical algorithm'. Rather, they will review the complete specification, including the detailed description of the invention, any specific embodiments that have been disclosed..." Hence, while Applicant disagrees that claims 1 and 2 merely recite a mathematical algorithm, this in itself is not a proper ground for rejection of the claims.

Applicant has amended the claims to further clarify that which is being claimed and now recites in amended claim 1:

> A method of restructuring a binary decision diagram representation of a hardware system, comprising acts of:
>
> arranging variables of the binary decision diagram in a representation of a graph, corresponding to the hardware system, the graph having a top, nodes and leaves, the nodes being labeled with the variables of the binary decision diagram and the leaves being labeled with a set of functions, thereby to generate labels for the nodes and leaves, whereby the set of functions labeling the leaves reachable from one of the nodes corresponds to the set of functions that depend on the variables labeling the one of the nodes;
>
> traversing the graph from the top down to produce a list of labels in a selected order; and
>
> using the list to restructure the binary decision diagram representation of the hardware system.

Amended claim 2 has also been amended to recite:

> An apparatus for restructuring a binary decision diagram representation of a hardware system, comprising :
>
> a first storage circuit to store first bits representing variables of the binary decision diagram;

a second storage circuit; and

a processor, coupled to the first storage circuit and the second storage circuit, adapted to arrange the variables of the binary decision diagram in a representation of a graph having a top, nodes, and leaves, to generate labels for the nodes and leaves, the nodes being labeled with the variables of the binary decision diagram and the leaves being labeled with a set of functions, the set of functions labeling the leaves reachable from one of the nodes corresponding to the set of functions which depend on the variables labeling the one of the nodes, the processor also being adapted to traverse the graph from the top down, and to output to the second storage circuit a list of labels in a selected order based upon traversal of the graph.

The MPEP states, e.g. at §2106(II)(A) that "only when the claim is devoid of any limitation to a practical application in the technological arts should it be rejected under 35 U.S.C. §101. Since Applicant's amended claims clearly recite their applicability to at least a hardware system, the rejection under 35 U.S.C. §101 should be withdrawn.

Claim Rejections Under 35 U.S.C. §102

At paragraph 17 of the Office Action, claims 1 through 11 where rejected under 35 U.S.C. §102(b) as being anticipated by Rudell and were further rejected under 35 U.S.C. 102(e) as being anticipated by Ashar. Applicant respectfully traverses these rejections as follows.

Applicant's Claims Distinguish Over Rudell

In the Office Action at paragraph 17, the Examiner states that Rudell discloses the use of dynamic variable ordering (DVO) as well as sifting algorithms for use with dynamic variable ordering and includes the heuristic order based on depth-first in the sifting algorithm. The Office Action further states that Rudell discloses sifting the variables in an uphill and pair wise swap based on position, and that the sifting may be run on a SPARC station that reads in the initial BDD from memory and outputs the optimized ordered BDD to memory.

Rudell discloses a dynamic variable ordering technique using a sifting algorithm. The variables in Rudell are moved dynamically to achieve dynamic ordering by the sifting. The order in which the variables are selected for sifting is determined according to which variable labels the greatest number of nodes. The sifting process in Rudell comprises exchanging a variable with its successor variable until the variable is sifted to the bottom of the DAG and then exchanging the variable with its predecessor until the variable is sifted up to the top of the DAG.

The variable is then moved to an optimum position which is the position with the smallest DAG size. The next variable is then sifted similarly and so on until all the variables have been sifted.

The disclosure of Rudell is essentially summarized in Applicant's description at p. 8, line 7-14 of the original application in the section titled "Description of the Prior Art." However, Rudell does not disclose or suggest the use of a pre-sifting or static ordering step as claimed in the present application. That is, Rudell only discloses dynamic variable ordering (DVO) techniques. Applicant's static variable ordering produces a list of variables which determines the order in which the variables are to be sifted. Applicant's claimed method produces a list of labels which is the order in which the variables are to be subsequently sifted. While Rudell requires moving the variables in traversing the graph, the Applicant's claims are not so limited.

Since Rudell does not disclose or suggest Applicant's recited method and apparatus for restructuring a binary decision diagram representation of a hardware system and traversing a graph from the top down to produce a list of labels in a selected order, the rejection is improper and should be withdrawn.

Applicant's Claims Distinguish Over Ashar

The Office Action rejected claims 1-11 under 35 U.S.C. §102(e) in view of Ashar. Office Action at paragraph 17. The Office Action states that Ashar discloses the use of sifting, breadth first, and depth first ordering, DVO, multiplexer circuitry model and using partitioning based on signal relationships, functional relationships or flow relationships on a SPARC. Applicants respectfully traverse this rejection as the claims patentably distinguish over the Ashar reference.

Ashar discloses computer code for simulating circuits to produce the same output responses as the circuit without having to fabricate the circuit itself. Ashar at col. 1 line 15. The simulations are derived from a BDD-based characteristic function representation of the circuit. Ashar discloses generating assembly code from the BDD of a single output function of a BDD-based characteristic function of a multiple output function for the purpose of functional simulation where each BDD node corresponds to an input variable and is translated to a sequence of machine instructions. Ashar at col. 2, line 40. Ashar further discloses, at col. 4, line 1 that the simulated circuit undergoes conventional output ordering, partitioning and BDD generation. The BDD-based characteristic function of Ashar is translated directly into and if-

then-else program. Ashar at col. 4, line 15. Ashar discloses at col. 6, line 12 that the implementation in Ashar uses a dynamic variable ordering (DVO) framework. Ashar references the work of Rudell (above) and states that dynamic variable ordering reorders the variables while building the BDD's. Variable sifting is used in Ashar and is according to a sifting algorithm which checks a matrix prior to each variable SWAP. The various techniques for determining when to stop swaping the variables are disclosed as well.

Ashar largely builds on the work of Rudell, and thus as discussed above with regard to Rudell, the Ashar reference discloses only dynamic variable ordering (DVO) and the sifting is determined according to which variable labels the greatest number of nodes.

Since Ashar does not disclose or suggest Applicant's recited method and apparatus for sifting variables in a binary decision diagram representation of a hardware system and traversing a graph from the top down, thereby to produce a list of labels in a selected order, the rejection is improper and should be withdrawn.

As neither Rudell nor Ashar disclose or suggest the recited method and apparatus claimed by the Applicant, Applicant's amended claims patentably distinguish over Rudell and Ashar, alone or in combination. Applicant therefore respectfully requests that the claim rejections of all independent claims under 35 U.S.C. §102 be withdrawn and the amended claims be allowed. Similarly, the dependent claims are allowable over the cited references for at least the same reasons as those given above regarding the independent claims. Accordingly, Applicant respectfully requests that the dependent claims likewise be allowed.

CONCLUSION

In view of the foregoing amendments and remarks, this application should now be in condition for allowance. A notice to this effect is respectfully requested. If the Examiner believes, after this amendment, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Respectfully submitted
Geoff Barrett, Applicant

By: Ibrahim M. Hallaj

Ibrahim Hallaj, Reg. No. 51,768
Robert A. Skrivanek, Jr., Reg No. 41,316
Wolf, Greenfield & Sacks, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210-2211
Tel. No.: (617) 720-3500
Attorney for Applicant

Docket No.: S01022.80126.US
Date: November 6, 2002
x11/06/02x

MARKED-UP CLAIMS

The following claims have been amended as shown below with underlining indicating additions and bracketing indicating deletions.

1. A method [for selecting an order in which to sift variables in]of restructuring a binary decision diagram representation of a hardware system, comprising acts of:[-]

arranging [the]variables of the binary decision diagram[on the nodes] in a representation of a graph[in which the], corresponding to the hardware system, the graph having a top, nodes and leaves, the nodes [are labelled]being labeled with the variables[of the system such that the set] of the binary decision diagram and the leaves being labeled with a set of functions[labelling], thereby to generate labels for the nodes and leaves, whereby the set of functions labeling the leaves reachable from [a node correspond]one of the nodes corresponds to the set of functions [which]that depend on the variables [labelling]labeling the [node]one of the nodes;[and]

traversing the graph [in a depth first manner, thereby]from the top down to produce a list of [said]labels in [said]a selected order[.]; and

using the list to restructure the binary decision diagram representation of the hardware system.

2. [Apparatus for selecting an order in which to sift variables in]An apparatus for restructuring a binary decision diagram representation of a hardware system, comprising :

a first storage circuit to store [storing]first bits representing [the]variables of the binary decision diagram;

a second [store]storage circuit; and

a processor, coupled to the first storage circuit and the second storage circuit, adapted to arrange the [said]variables of [said]the binary decision diagram in a representation of [the nodes of]a graph [in which the]having a top, nodes[are labelled], and leaves, to generate labels for the nodes and leaves, the nodes being labeled with the variables [such that the]of the binary decision diagram and the leaves being labeled with a set [of]of functions[labelling], the set of functions labeling the leaves reachable from [a node corresponds]one of the nodes corresponding to the set of functions which depend on the variables [labelling]labeling the [node; and]one of the nodes,

the processor also being adapted to traverse the graph [in a depth-first manner such that said processor outputs]from the top down, and to [said]output to the second [store]storage circuit a list of [said]labels in [said]a selected order based upon traversal of the graph.

3. A method [for]of restructuring a binary decision diagram representative of a hardware system, the binary decision diagram including a plurality of variables, the method comprising acts of:[-]

arranging [the]variables of the binary decision diagram[on the nodes] in a representation of a graph[in which the] corresponding to the hardware system, the graph having a top, nodes[are labelled], and leaves, to generate labels for the nodes and leaves, the nodes being labeled with the variables[of the system such that the set] of the binary decision diagram and the leaves being labeled with a set of functions[labelling], the set of functions labeling the leaves reachable from [a node corresponds]one of the nodes corresponding to the set of functions which depend on the variables [labelling]labeling the [node]one of the nodes;[and]

traversing the graph [in a depth-first manner]from the top down to produce a list of [said]the labels in a selected order;

[using said] sifting the variables based on the selected order[, controlling sifting each variable.]; and

restructuring the binary decision diagram based on the act of sifting the variables.

4. [A]The method [as claimed in]of claim [3]3, wherein [said]the variables are sifted one-by-one to a deepest[best] location.

5. [A]The method [as claimed in]of claim [3]3, wherein [said]the variables are sifted one-by-one [is]in [said]the selected order to a deepest [best]location followed by sifting in reverse order to a shallowest[best] location.

6. [Apparatus]An apparatus for restructuring a binary decision diagram representative of a hardware system, comprising:[-]

storage circuitry for storing bits representative of a set of functions as a binary decision [diagrams]diagram corresponding to the hardware system, the binary decision diagram having a

plurality of nodes[labelled by], the nodes being labeled with variables to provide labels for the nodes; and

a processor [for detecting]adapted to detect a number of nodes of [said]the binary decision diagram, and in response to [said]the detection,

arranging the variables of [said]the binary decision diagram on [the nodes of]a graph [in which the]having a top, nodes [are labelled such that]and leaves, to generate labels for the [set]nodes and leaves, the nodes being labeled with the variables of the binary decision diagram and the leaves being labeled with a set of functions[labelling], the set of functions labeling the leaves reachable from [a node corresponds]one of the nodes corresponding to the set of functions which depend on the variables [labelling]labeling the [node]one of the nodes,

traversing the graph [in a depth-first fashion]from the top down, to produce a list of the labels in a selected order, and[using said selected order, controlling]

sifting [of]the variables of [said]the binary decision [diagrams;]diagram based on the selected order, wherein [said]the sifted [binary decision diagram is]variables are written by [said]the [processor]processors to [said]the storage [circuits]circuitry.

7. A method for proving the properties of a hardware system, the hardware system comprising a plurality of internal signals, wherein a plurality of functions determine variables of the internal signals, the method comprising acts of:[-]

representing [said]the hardware system as a binary decision [diagrams]diagram having a plurality of nodes[labelled by], the nodes being labeled with variables to provide labels for the nodes;

substituting the functions which determine the variables of the internal signals;

arranging the variables of [said]the binary decision diagram on [the nodes of]a graph [in which]having a top, nodes and leaves, the nodes [are]being [labelled]labeled with the variables of the system [such that]and the leaves being labeled with a set of functions [labelling]to generate labels for the nodes and leaves, whereby the set of functions labeling the leaves reachable from [a node]one of the nodes corresponds to the set of functions which depend on the variables [labelling]labeling the [node]one of the nodes;[and]

traversing the graph [in a depth-first manner]from the top down to produce a list of [said]the labels in a selected order; and

[using said selected order, controlling sifting each variable.]

sifting the variables of the binary decision diagram based on the selected order.

8. [Apparatus]An apparatus for proving [the]properties of a hardware system, the hardware system comprising a plurality of internal signals, wherein a plurality of functions determine values of the internal signals, the apparatus comprising:

storage circuitry for storing bits representative of a set of functions which represent the hardware system as a binary decision [diagrams]diagram having a plurality of nodes[labelled by], the nodes being labeled with variables to provide labels for the nodes;

a processor [for substituting]that substitutes the functions which determine the values of the internal signals into the set of functions representing [said]the system and [detecting]detects an increase in [the]a number of the nodes of [said]the binary decision diagram, and[,] in response to [such]the detection[arranging the variable of said], arranges the variables of the binary decision diagram on [the nodes of]a graph [in which]having a top, nodes and leaves, the nodes [are]being [labelled]labeled with [the variables of the system such that the]a set of functions[labelling] to generate labels for the nodes and leaves, whereby the set of functions labeling the leaves reachable from [a node corresponding]one of the nodes corresponds to the set of functions which depend on the variables [labelling]labeling the [node, traversing the graph in a depth-first fashion to produce a list]one of the nodes,traverses the labels in [said]a selected order[,] and [using said selected order controlling sifting of]sifts the variables of [said]the binary decision diagram based on the selected order; and

[further comprising] a second store, wherein [said sifting]the sifted variables of the binary decision diagram [is]are written by [said]the processor to [said]the second store.

9. [Apparatus as claimed in]The apparatus of claim [8]8, wherein [said]the number is a threshold derived from an original number of nodes.

10. Apparatus as claimed in claim [8]8, wherein [said number of nodes is]the number is a number of nodes which [branches]branch on a predetermined variable.

11. [Apparatus as claimed in]The apparatus of claim [8]8, wherein [said]the number is an absolute number.

METHOD AND APPARATUS FOR [PROVING SYSTEM PROPERTIES]RESTRUCTURING A BINARY DECISION DIAGRAM

BACKGROUND OF INVENTION

FIELD OF INVENTION

The present invention relates to a method and apparatus for reducing the complexity of a representation of a hardware system.

DESCRIPTION OF THE PRIOR ART

The first stage in synthesizing and proving the properties of a system is a compilation process in which the system is [modelled]modeled by representation as a set of functions comprising[:-] a first subset of functions which determines the value of system outputs as a function of system inputs, system states represented by state bits, and internal signals; a second subset of functions which determines the values of state bits on the next clock cycle as a function of system inputs, system states represented by state bits, and internal signals; and a third subset of functions which determines the values of internal signals as a function of system inputs, system states, and internal signals.

To enable or accelerate formal proof of the system and its properties, internal signals may be eliminated from the system model by substituting them into the functions which refer to them. In the course of this substitution, the representation of the model may become extremely large. If this occurs, it is possible to detect an explosion in the size of the representation and to suspend the substitution process while restructuring the representation to seek a reduction in size.

Typically in a compilation process, static relationships between signals in the system model can be destroyed by dynamic restructuring operations. This can lead to a further explosion later during the substitution process.

It would be advantageous to take static relationships into account during the dynamic restructuring process.

One technique of representing functions and internal signals is by the use of binary decision diagrams (BDD's). A binary decision diagram is a representation of a digital function which contains the information necessary to implement the function. The diagram is a tree-like structure having a root and plural nodes, where the root represents the digital function and the nodes are [labelled]labeled with variables. Each node has two branches, one representing the assertion that the variable [labelling]labeling the node is 1, and the other representing the assertion that the variable [labelling]labeling the node is 0. In a BDD, "ordering" relates to the order in which variable names are encountered during traversal of the graph. Better orderings result in fewer nodes in the graph.

SUMMARY OF INVENTION

According to a first aspect of the present invention, there is provided a method for selecting an order in which to sift variables in a binary decision diagram comprising[:-] arranging the variables of the binary decision diagram on the nodes of a graph in which the nodes are [labelled]labeled with the variables of the system such that the set of functions [labelling]labeling leaves reachable from a node[,] correspond to the set of functions which depend on the variables [labelling]labeling the node; and traversing the graph in a depth first manner, thereby to produce a list of [said]the labels in [said]the selected order.

According to a second aspect of the present invention there is provided an apparatus for selecting an order in which to sift variables in a binary decision diagram comprising a first store storing bits representing the variables of the binary decision diagram; a second store; and a processor adapted to arrange the [said]variables of [said]the binary decision diagram in a representation of the nodes of a graph in which the nodes are [labelled]labeled with the variables such that the set of functions [labelling]labeling leaves reachable from a node corresponds to the set of functions which depend on the variables [labelling]labeling the node; and to traverse the graph in a depth-first manner such that [said]the processor[means] outputs to [said]the second store a list of [said]labels in [said]the selected order.

According to a third aspect of the present invention there is provided a method for restructuring a binary decision diagram representative of a hardware system, comprising[:-] arranging [the]variables of the binary decision diagram on [the]nodes of a graph in which the nodes are [labelled]labeled with the variables of the system such that the set of functions [labelling]labeling leaves reachable from a node corresponds to the set of functions which depend on the variables [labelling]labeling the node; and traversing the graph in a depth-first manner to produce a list of [said]labels in a selected order; using [said]the selected order, controlling sifting of each variable.

Preferably [said]the variables are sifted one-by-one to a deepest best location. Advantageously [said]the variables are sifted one-by-one in [said]the selected order to a deepest best location followed by sifting in reverse order to a shallowest best location.

According to a fourth aspect of the present invention there is provided apparatus for restructuring a binary decision diagram comprising[:-] storage circuitry for storing bits representative of a set of functions as [a]binary decision diagrams having a plurality of nodes [labelled]labeled by variables; a processor for detecting a number of nodes of [said]the binary decision diagram, and in response to such detection, arranging the variables of [said]the binary decision diagram on the nodes of a graph in which the nodes are [labelled]labeled such that the set of functions [labelling]labeling leaves reachable from a node corresponds to the set of functions which depend on the variables [labelling]labeling the node, traversing the graph in a depth-first fashion to produce a list of labels in a selected order and using [said]the selected order, controlling sifting of the variables of [said]the binary decision diagrams; wherein [said]the sifted binary decision diagram is written by [said]the processor to [said]the storage circuitry.

According to a fifth aspect of the present invention there is provided a method for proving properties of a hardware system comprising[:-] representing [said]the system as binary decision diagrams having a plurality of nodes [labelled]labeled by variables; substituting [the]functions which determine [the]variables of internal signals; arranging the variables of a binary decision diagram on the nodes of a graph

in which the nodes are [labelled]labeled with the variables of the system such that the set of functions [labelling]labeling leaves reachable from a node corresponds to the set of functions which depend on the variables [labelling]labeling the node; [and]traversing the graph in a depth-first manner to produce a list of [said]the labels in a selected order; and using [said]the selected order, controlling sifting of each variable.

According to a sixth aspect of the present invention there is provided apparatus for proving properties of a hardware system comprising[:] storage circuitry for storing bits representative of a set of functions which represent the hardware system as binary decision diagrams having a plurality of nodes [labelled]labeled by variables; processor means for substituting functions which determine[the] values of internal signals into the set of functions representing [said]the system and detecting an increase in the number of nodes of [said]the binary decision diagram, and, in response to such detection arranging the variable of [said]the binary decision diagram on the nodes of a graph in which the nodes are [labelled]labeled with the variables of the system such that the set of functions [labelling]labeling leaves reachable from a node corresponding to the set of functions which depend on the variables [labelling]labeling the node, traversing the graph in a depth-first fashion to produce a list of labels in [said]the selected order, and using [said]the selected order controlling sifting of the variables of [said]the binary decision diagram; and further comprising a second store, wherein [said]the [sifting]sifted binary decision diagram is written by [said]the processor means to [said]the second store.

Preferably [said]the number is a threshold derived from an original number of nodes. Alternatively [said]the number is the number of nodes which branches on a predetermined variable. Alternatively [said]the number is an absolute number.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the present invention will now be described with respect to the following drawings in which:[-]

Figure 1 shows a binary decision diagram for the function

f = x OR y;

Figure 2 shows a logical diagram of a multiplexer;

Figure 3 shows a binary decision diagram for the equation[;]

$$\mathbf{b_i = NOT\ (a_i\ AND\ s_i);}$$

Figure 4 shows the binary decision diagram for the equation

$$\mathbf{d = NOT\ (b_1\ AND\ b_2\ AND\ b_3 ... b_n);}$$

Figure 5 is an optimally ordered substitution of the equations of Figures 3 and 4;

Figure 6 shows a graph of relationships between the variables of the multiplexer of Figure 2;[and]

[Figurer]Figure 7 shows an example of apparatus arranged to implement an embodiment of the present invention; and

Figure 8 shows a flow diagram indicating the features of an embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

A Binary Decision Diagram (hereinafter referred to as a [BBD]BDD) is a directed acyclic graph representative of a Boolean function as a decision procedure based on the variables on which it depends. For instance, for the function:

f = x OR y,

f can be implemented by the decision procedure "if x then true else if y then true else false". Each of the "if...then...else..." constructs of this decision procedure can be represented as a node in a graph.

Referring to Figure 1, the first node 1 is [labelled]labeled with the variable x and there are two branches from this first node, one 11 is "true" and the other 12 is "if y then true else false". This other branch 12 leads to a second node 2 which is [labelled]labeled with the variable y, which in turn has two branches 21, 22 of which one is "true" and the other is "false".

It will be understood that although the nodes 1 and 2 are described above as being [labelled]labeled with variables, nevertheless these labels could in fact refer to functions which upon evaluation would give rise to the logical values "true" or "false".

Referring now to Figure 2, a multiplexer consists of a first set of n NAND gates 10_1-10_n, each gate having two respective inputs a_1-a_n, s_1-s_n. The outputs, lines b_1 and b_n of the gates, are connected to an n-input NAND gate 20 having an output d.

Thus, in terms of a system as described in the preamble to this patent application, the multiplexer of Figure 2 has system inputs (a_1-a_n, s_1-s_n), internal signals (b_1-b_n) and a system output (d). The output d is related to the internal signals b_1-b_n by the equation:[-]

$$\mathbf{d = NOT\ (b_1\ AND\ b_2\ AND\ b_3\ \ldots b_n),}$$

and each internal signal b_i to the respective inputs a_i and s_i by the equation:

$$\mathbf{b_i = NOT\ (a_i\ AND\ s_i),}$$

Thus,

$$\mathbf{d = (a_1\ AND\ s_1)\ OR\ (a_2\ AND\ s_2)\ OR\ \ldots (a_n\ AND\ s_n),}$$

Referring to Figure 3 the relationship b_i = NOT (a_i AND s_i) is shown as a binary decision diagram.

Figure 4 shows the binary decision diagram representation of the expression for d in terms of the internal signals b.

By inspection, there are 3n variables (a_i, s_i and b_i) and there are thus (3n) ! apparently equally good orderings possible. However, by inspection of the overall equation for the device it would be seen that a_1 and s_1 are associated together, a_2 and s_2 are associated together and so on which means that there are in fact only n! orderings which are optimal for the entire system.

An advantage of the present invention is that it enables more information about the system as a whole to be taken into account when performing operations which would otherwise not take this information into account. Failing to take the information into account can result in following paths which do not lead to a solution, or which are highly inefficient in reaching the solution.

Figure 5 shows a binary decision diagram for the multiplexer of Figure 2 in which the respective pairs of inputs are associated together.

The size of a binary decision diagram is sensitive to the order in which the variables are inspected, and efficient BDD reordering is very important. One algorithm for reordering is "sifting", wherein each variable is taken in turn and the

best position of it is found by trying it in every possible position of the BDD. It is then necessary to decide which variable to take first. A known and frequently successful tool for doing this is to rank the variables according to which variable labels the greatest number of nodes and then to sift in the order of ranking.

In the present BDD, it is clear that each variable labels a single node and thus it would not be possible using known techniques to identify a highest ranking variable. Conventionally, in such a situation, an arbitrary order for sifting would be used.

The present invention makes use of a function graph which is traversed to determined an order for sifting.

As used herein, a function graph is a directed acyclic graph where the leaves are [labelled]labeled with functions and the nodes are [labelled]labeled with sets of variables (non-empty). The only restriction put on this graph is that a variable which is in the set [labelling]labeling a node is in the "cone" of all the functions at the leaves below it and no others. This restriction plus the fact that the sets of variables must be non-empty, is enough to ensure that the graph is unique. The "cone" of a function is herein defined to be all those variables on which a function depends, either directly or through the intermediate signals on which it depends.

Using a function graph to define an ordering of the variables in a BDD to [minimise]minimize its size may be related to the register allocation technique used in software compilation in that the ordering of the variables is derived from a traversal of the function graph in such a way that no node is visited before all of its predecessors has been visited, but each node is visited as soon as all its predecessors have been visited, unless there is a race between more than one node, in which case one of the competing nodes is chosen and its subgraphs traversed first.

Figure 6 shows a function graph for the multiplexer of Figure 2 having a root [labelled]labeled by b_1.b_n intermediate nodes [labelled]labeled by a_1, s_1, a_2, s_2 ... a_n, s_n and leaves as shown. Traversing this function graph from the top down gives the order:[-]

b_1, b_2...b_n a_1, s_1, a_2, s_2 ... a_n, s_n.

By using this order which is derived from static information of the system, the binary decision diagrams of (in this case) Figures 1, 3 and 4 are sifted to provide an optimal order. This order is that represented by Figure 5.

It should be noted that substitution may be effected without restructuring the BDD, while monitoring the size of the BDD. If an explosion in BDD is detected, sifting is then effected on the basis of the order provided by the present invention.

Figure 7 shows an example of apparatus to implement an embodiment of the present invention. A first storage circuitry 30 stores data representative of the variables of a binary decision diagram.

The first storage circuitry has an input 32 for receiving the variables. The output 34 of the first storage circuitry is coupled to an input 42 of a processor 40. The processor receives the variables via the output 34 of the first storage circuitry and consults a function graph by arranging the variables in a representation of the nodes of the graph such that the nodes are [labelled]labeled with the variables so that the set of functions [labelling]labeling leaves reachable from a node corresponds to the set of functions which depend on the variables [labelling]labeling the node, as shown in Figure 6. The processor then traverses the graph in a depth-first manner, as indicated in Figure 6 by the arrow, to construct a list of the labels in a selected order. The list is output from the processor via output 44 which is coupled the input 52 of second storage circuitry.

Figure 8 illustrates an exemplary process for restructuring a BDD that represents a hardware system as described above. In act 1000, the hardware system is represented as a set of functions that determine values of system variables (input signals, internal signals, output signals). In act 1002, the set of functions is represented as binary decision diagrams having a plurality of nodes labeled by the variables of the system. In act 1004, functions that determine the value of internal signals in the system are substituted into the set of functions representing the system. When an increase in a number of nodes in the binary decision diagram is detected in act 1006, wherein the number is a threshold derived from an original number of nodes, the process proceeds to act 1008. In act 1008, in response to the detection at act 1006, processor 40 arranges the variables of the binary decision diagram on the

nodes of a graph whereby the nodes are labeled by the variables. In act 1010, the graph is traversed from the top down to produce a list of the labels in a selected order, and in act 1012, the variables are sifted in the selected order to reduce the size of the binary decision diagram. In act 1012, the variables are sifted one-by-one to a deepest best location and then sifted in reverse order to a shallowest best location.

While the invention has been previously shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

CLAIMS

[What is claimed is:]

1. A method for selecting an order in which to sift variables in a binary decision diagram comprising:-

arranging the variables of the binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node correspond to the set of functions which depend on the variables labelling the node; and

traversing the graph in a depth first manner, thereby to produce a list of said labels in said selected order.

2. Apparatus for selecting an order in which to sift variables in a binary decision diagram comprising a first store storing bits representing the variables of the binary decision diagram;

a second store; and

a processor adapted to arrange the said variables of said binary decision diagram in a representation of the nodes of a graph in which the nodes are labelled with the variables such that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node; and to traverse the graph in a depth-first manner such that said processor outputs to said second store a list of said labels in said selected order.

3. A method for restructuring a binary decision diagram representative of a hardware system comprising:-

arranging the variables of the binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node; and

traversing the graph in a depth-first manner to produce a list of said labels in a selected order;

using said selected order, controlling sifting each variable.

4. A method as claimed in claim 3 wherein said variables are sifted one-by-one to a deepest best location.

5. A method as claimed in claim 3 wherein said variables are sifted one-by-one is said selected order to a deepest best location followed by sifting in reverse order to a shallowest best location.

6. Apparatus for restructuring a binary decision diagram comprising:-

storage circuitry for storing bits representative of a set of functions as binary decision diagrams having a plurality of nodes labelled by variables;

a processor for detecting a number of nodes of said binary decision diagram, and in response to said detection, arranging the variables of said binary decision diagram on the nodes of a graph in which the nodes are labelled such that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node, traversing the graph in a depth-first fashion to produce a list of labels in a selected order and using said selected order, controlling sifting of variables of said binary decision diagrams;

wherein said sifted binary decision diagram is written by said processor to said storage circuits.

7. A method for proving the properties of a hardware system comprising:-

representing said system as binary decision diagrams having a plurality of nodes labelled by variables;

substituting the functions which determine the variables of internal signals;

arranging the variables of said binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node corresponds to the set of functions which depend on the variables labelling the node; and

traversing the graph in a depth-first manner to produce a list of said labels in a selected order;

using said selected order, controlling sifting each variable.

8. Apparatus for proving the properties of a hardware system comprising:

storage circuitry for storing bits representative of a set of functions which represent the hardware system as binary decision diagrams having a plurality of nodes labelled by variables;

a processor for substituting functions which determine the values of the internal signals into the set of functions representing said system and detecting an increase in the number of nodes of said binary decision diagram, and, in response to such detection arranging the variable of said binary decision diagram on the nodes of a graph in which the nodes are labelled with the variables of the system such that the set of functions labelling leaves reachable from a node corresponding to the set of functions which depend on the variables labelling the node, traversing the graph in a depth-first fashion to produce a list of labels in said selected order, and using said selected order controlling sifting of the variables of said binary decision diagram; and

further comprising a second store, wherein said sifting binary decision diagram is written by said processor to said second store.

9. Apparatus as claimed in claim 8 wherein said number is a threshold derived from an original number of nodes.

10. Apparatus as claimed in claim 8 wherein said number of nodes is the number of nodes which branches on a predetermined variable.

11. Apparatus as claimed in claim 8 wherein said number is an absolute number.

ABSTRACT

A method for selecting an order in which to sift variables in a binary decision diagram[by]. The method includes an act of arranging the variables of [a]the binary decision diagram on [the]nodes of a graph, with the nodes of the graph being [labelled]labeled with the variables of the system such that [the]a set of functions [labelling]labeling the leaves reachable from a node correspond to the set of functions which depend on the variables [labelling]labeling the node[, and]. The method further includes an act of traversing the graph in a depth first manner[, thereby] to produce a list of the labels in the selected order.

[Fig. 1]